

# Ruby入門

高宮 安仁 @yasuhito



# 品詞+文法





# 登場する品詞

- キーワード
- 名詞
- 動詞

# キーワード(予約語)

alias and BEGIN begin break case  
class def defined do else elsif  
END end ensure false for if in  
module next nil not or redo  
rescue retry return self super  
then true undef unless until  
when while yield

```
class HelloTrema < Trema::Controller  
  def start(_args)  
    logger.info 'Trema started.'  
  end  
end
```

- ・ キーワードは構造を決める

```
class HelloTrema < Trema::Controller
  def start(_args)
    logger.info 'Trema started.'
  end
end
```

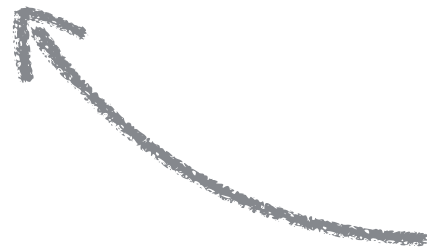
- ・ クラス定義 (class ... end)
- ・ 子クラス < 親クラス名

# 固有名詞=定数

Tsutenkaku

Rainbow Bridge

HelloTrema



- 大文字で始まる
- 内容を変更できない

```
class HelloTrema < Trema::Controller
  def start(_args)
    logger.info 'Trema started.'
  end
end
```

- ・定数は大文字で始まる



# メソッド＝動詞

- boy.run
- girl.write 'abc'
- logger.info

Flashcards

Verbs 1

©www.kids-pages.com



walk



run



play



sleep



read



write



```
class HelloTrema < Trema::Controller
  def start(_args)
    logger.info 'Trema started.'
  end
end
```

- `logger` = ログ
- `info` = `info` ログの出力

```
class HelloTrema < Trema::Controller
```

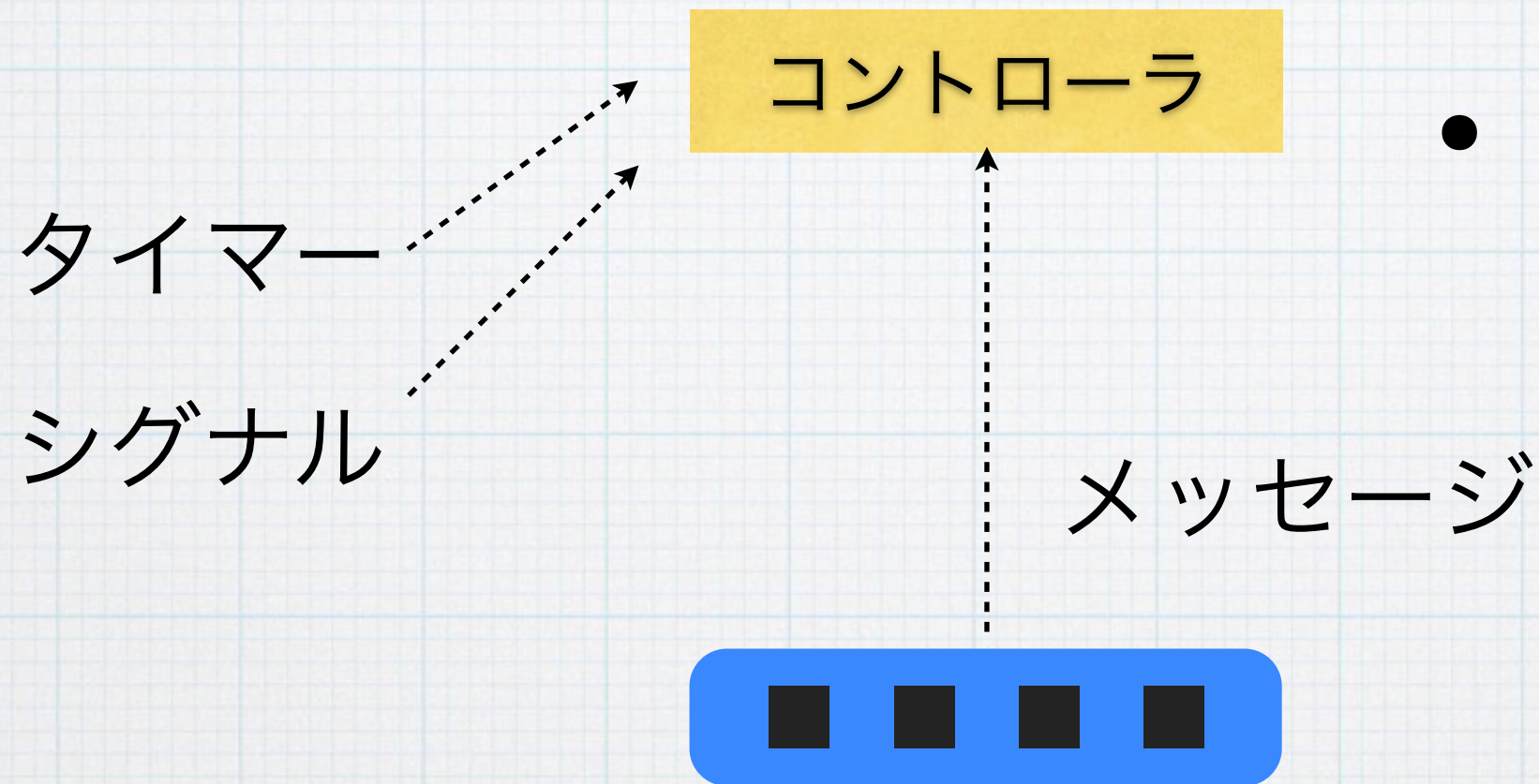
```
  def start(_args)
```

```
    logger.info 'Trema started.'
```

```
  end
```

```
end
```

- `def ... end` = メソッド定義
- `start` はハンドラメソッド



- ハンドラ1
- ハンドラ2
- ハンドラ3

コントローラの外界からのメッセージや  
イベントに応じ、コントローラの対応する  
ハンドラメソッドが呼ばれる

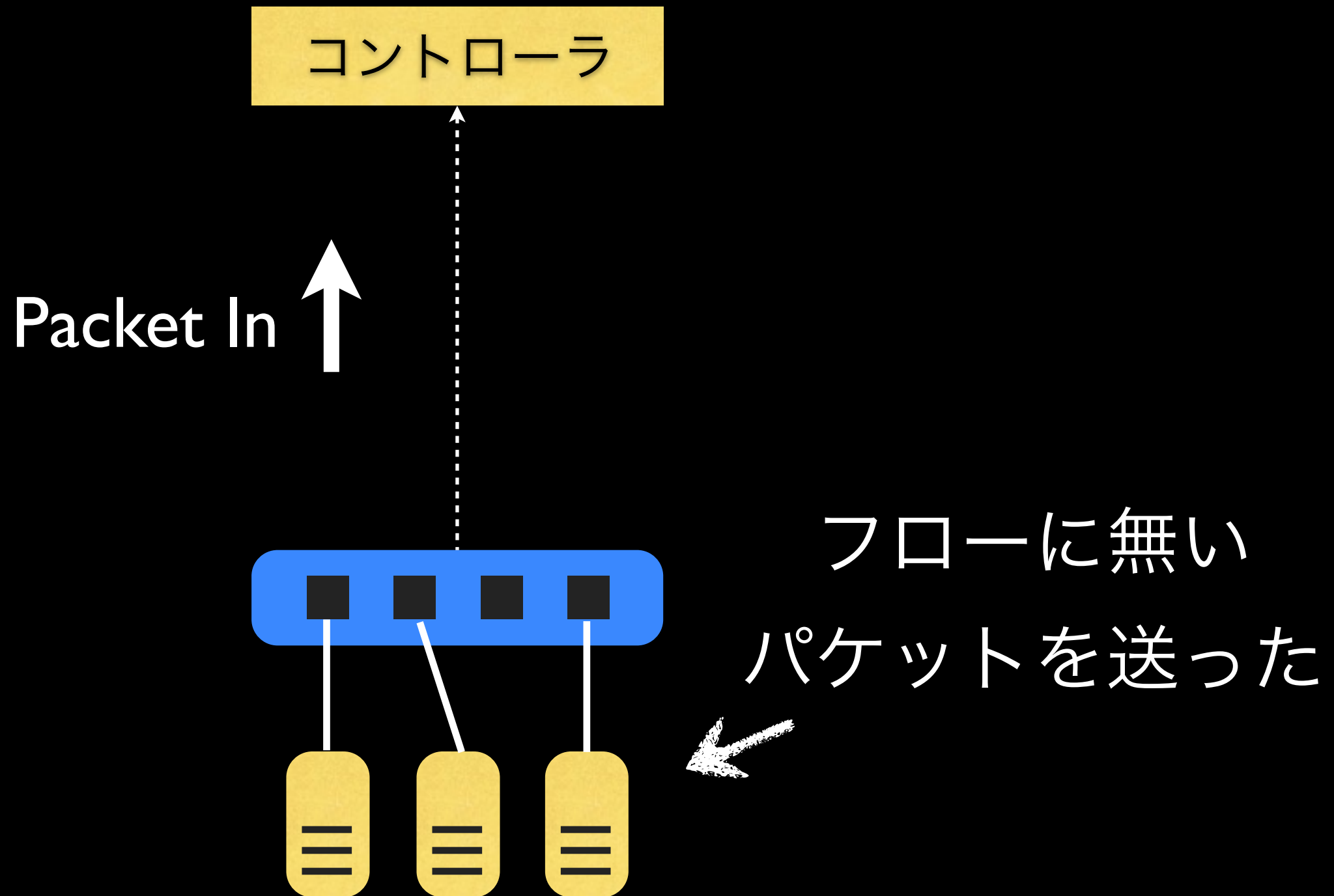
start() ← コントローラが

起動した

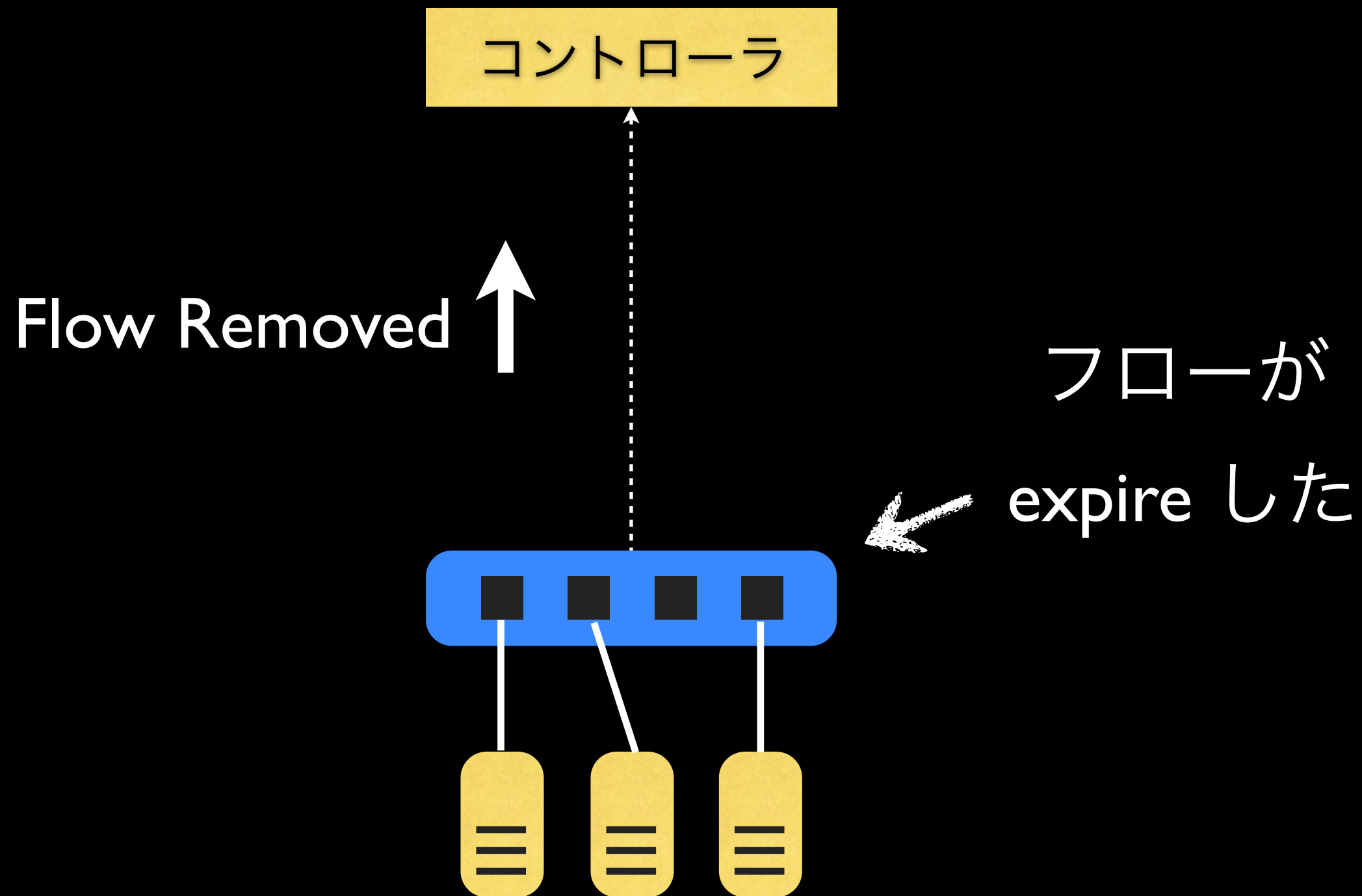
コントローラ



packet\_in()



flow\_removed()



# ハンズラー覧

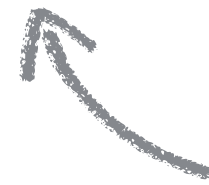
start switch\_ready  
switch\_disconnected  
packet\_in flow\_removed  
port\_status openflow\_error  
features\_reply stats\_reply  
barrier\_reply  
get\_config\_reply  
queue\_get\_config\_reply  
vendor

# 課題

- スイッチを停止したら

“Bye 0xabc”

と表示せよ



スイッチの**DPID**

- ・ 接続を  
捕捉

- ・ 切断を  
捕捉

```
class FooBar < Trema::Controller
  def switch_ready(dpid)
    # ...
  end

  def switch_disconnected(dpid)
    # ...
  end
end
```



```
% trema run foobar.rb -c network.conf
```



```
vswitch { dpid 0x1 }
```

```
vswitch { dpid 0x2 }
```

```
vswitch { dpid 0x3 }
```

```
% trema stop 0x1 # 仮想スイッチを殺す
```

```
% trema start 0x1 # 仮想スイッチを復活
```

- `trema stop/start` コマンドで仮想スイッチを操作