

チェリー本 第7章

selfキーワードとクラスの継承

WEF2105 宮崎ひとみ

selfキーワード

【日本語訳】

self => 自身の ⇒ Rubyでは、オブジェクト自身を指す！

例)

- ・ クラス内の selfキーワード = クラス自身
- ・ インスタンスメソッド内の selfキーワード = インスタンス自身
- ・ クラスメソッドの selfキーワード = クラス自身

クラス内のselfキーワード

```
class Taiyaki  
  puts self  
end
```

#=> **Taiyaki**
(クラス自身)



ちなみに… A.キーワードとは何？

Q.予約語とも呼ばれ、メソッドとは異なる。

nil、end、def、when、true など。

インスタンスメソッド内のselfキーワード

```
class Taiyaki  
  def hoge  
    puts self  
  end  
end
```

taiyaki = Taiyaki.new



taiyaki.hoge

#=> #<Taiyaki:0x00007f8524034f73>
(インスタンス自身)



クラスメソッド内のselfキーワード

```
class Taiyaki
  def self.fuga
    puts self
  end
end
```

Taiyaki.fuga #=> Taiyaki
(クラス自身)



2 行目

self.メソッド名 : クラスメソッドを定義

3 行目

puts self : クラスメソッド内の、
self を 出力する処理

分かりづらい部分：selfキーワードは省略されることが多い

```
class Taiyaki
  def hoge #インスタンスメソッド
  end

  def self.hoge #クラスメソッド
  end

  def fuga
    self.hoge #インスタンスメソッドのhogeが呼ばれる
    hoge      #      "      (selfは省略可)
    self.class.hoge #クラスメソッドのhogeが呼ばれる
  end

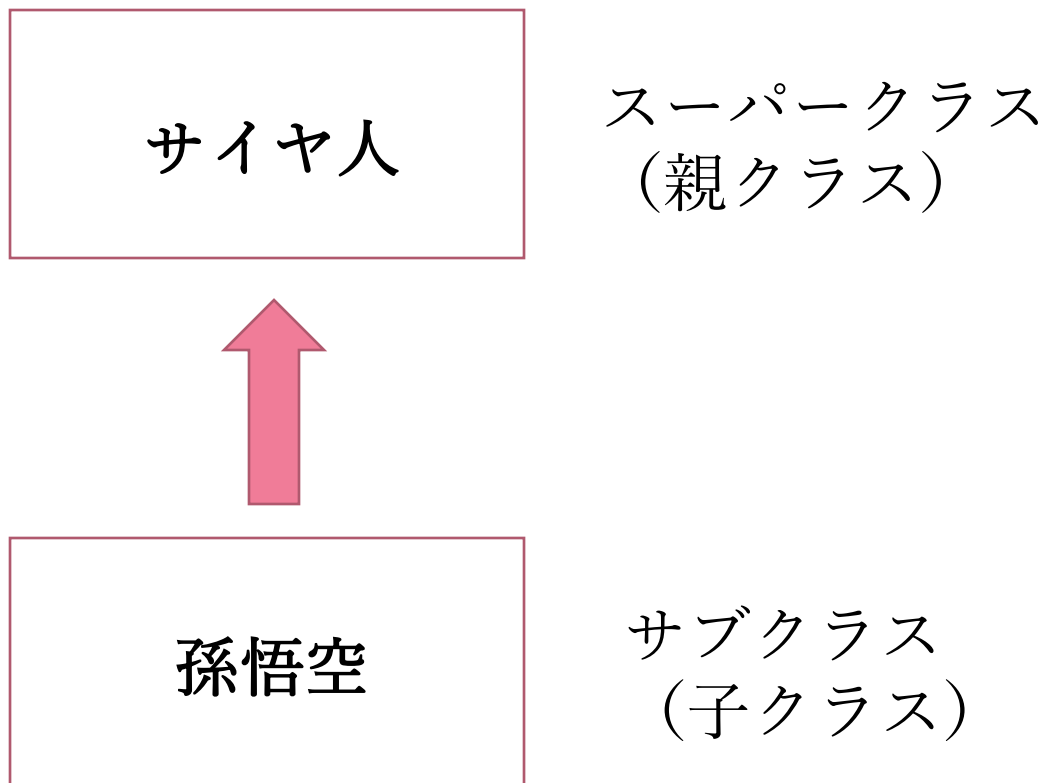
  def self.fuga
    self.hoge #クラスメソッドのhogeが呼ばれる
    hoge      #      "      (selfは省略可)
  end
end
```

参考

<https://qiita.com/suzuki-koya/items/1553c405bee73f83bbc>

クラスの継承

例)



継承とは？

元のクラスの要素を
受け継ぐこと

悟空クラスは、サイヤ人クラスを継承する

```
class Goku < SaiyaJin
  # 何も書かなくても...
end
```

継承



```
class SaiyaJin
  attr_reader :name, :place

  def initialize(name, place)
    @name = name
    @place = place
  end
end
```

```
goku = Goku.new('孫悟空', '地球')
```

```
goku.name => '孫悟空'
```

```
goku.place => '地球'
```


superで、スーパークラス(サイヤ人)のメソッドを呼べる

```
class Goku < SaiyaJin
  attr_reader :vehicle
```

乗り物を追加

```
  def initialize(name,place,vehicle)
    super(name,place)
```

```
    @vehicle = vehicle
```

```
  end
```

```
end
```

スーパークラスの
同名メソッドを呼び出す



```
class SaiyaJin
  attr_reader :name,:place
```

```
  def initialize(name,place)
    @name = name
    @place = place
```

```
  end
```

```
end
```

```
goku = Goku.new('孫悟空','地球','筋斗雲')
```

```
goku.name => '孫悟空'
```

```
goku.place => '地球'
```

```
goku.vehicle => '筋斗雲'
```

悟空クラスの技(メソッド)は、悟飯クラス(サブクラス)に受け継がれる

```
class Gohan < Goku
```

何も書かなくても...

```
end
```



```
class Goku < SaiyaJin  
  attr_reader :vehicle
```

```
  def initialize(name, place, vehicle)  
    super(name, place)  
    @vehicle = vehicle  
  end
```

```
  def kamehame_ha  
    puts '手からビーム'  
  end
```

```
end
```

カメハメ波を追加

```
gohan = Gohan.new('悟飯', '地球', '觔斗雲')
```

```
gohan.kamehame_ha => '手からビーム'
```

Kamehame_haメソッドのオーバーライド

```
class Gohan < Goku
```

目からビームを出したい
→ スーパークラスの同名メソッド内を
書き換える
(これをオーバーライドと呼ぶ)

```
def kamehame_ha  
  puts '目からビーム'  
end
```

```
end
```



```
class Goku < SaiyaJin  
  attr_reader :vehicle
```

```
  def initialize(name, place, vehicle)  
    super(name, place)  
    @vehicle = vehicle  
  end
```

```
  def kamehame_ha  
    puts '手からビーム'  
  end
```

```
end
```

gohan = Gohan.new('悟飯', '地球', '觔斗雲')

gohan.kamehame_ha => '目からビーム'

上書きされる

ご清聴ありがとうございました