

はじめよう／

ジブン 専用 パソコン

第41回 国産プログラミング言語 「ルビー」を使ってみよう

世の中には、数百とも数千ともいわれるプログラミング言語がある。なぜそんなんにたくさんあるのかというと、それぞれに得意なことがあるからだ。今回は、プログラミング言語の人気ランキング※で13位のRubyを取り上げるぞ。

※TIOBEインデックス：<https://www.tiobe.com/tiobe-index/>

ジブン専用パソコン特設サイト

[https://prog.kodomonokagaku.com/
category/raspberryPi](https://prog.kodomonokagaku.com/category/raspberryPi)

キットの情報は
KoKa Shop!へ
定期購読者特典割引あり
購入ページ



今回の内容は
普通のパソコンでもできるよ。

国産プログラミング言語「ルビー」

ルビーは1993年に日本のまつもとゆきひろさんが開発したプログラミング言語だ。TIOBEインデックス上位20位の中に入っている唯一の国産言語でもある。世界的に人気がある理由の1つは、Webアプリケーションの作成に向いているからだ。「クックパッド」や「価格.com」、「hulu」といったサイトでも使われているぞ。

ルビーは、オブジェクト指向型に分類される言語だ。実は、スクラッチもオブジェクト指向型言語なので、似ているところがあるよ。

ルビーのプログラムを書いてみよう

ジブン専用パソコンのOS、ラズビアン・ストレッヂには、ちょっと古いルビー(ruby 2.3.3p222)がインストールされている。他のパソコンを使って

ゲーム

工作

プログラミング



ラズベリーパイ
Raspberry Pi
大活用

```
1 self.when(:flag_clicked) do-
2   hide-
3   switch_costume("陰性")-
4   9.times do-
5     create_clone("_myself_")-
6   end-
7   switch_costume("陽性")-
8   1.times do-
9     create_clone("_myself_")-
10  end-
11 end-
12 ~
13 self.when(:start_as_a_clone) do-
14   show-
15 go_to("random")-
16 self.direction = rand(1..360)-
17 loop do-
18   move(20)-
19   bounce_if_on_edge-
20   if touching_color("#ff33ff")-
21     switch_costume("陽性")-
22   end-
23 end-
24 end-
25 end-
```

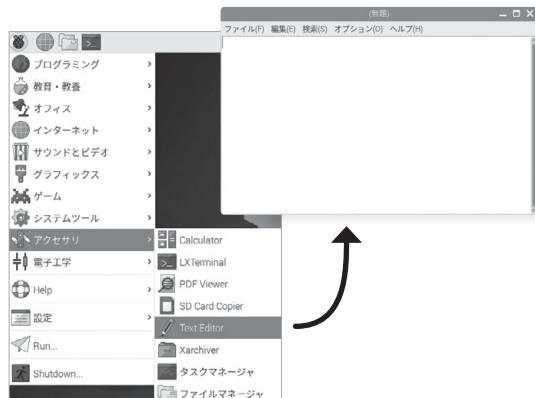
監修・原案／
青山学院大学大学院 特任教授
阿部和広
構成・文／塩野祐樹

いる人はルビーのサイトからインストールしてね。

ルビーのウェブサイト：

<https://www.ruby-lang.org/ja/>

ルビーはテキスト型のプログラミング言語なので、文字を使ってプログラムを書いていく。そのため使うのがテキストエディターだ。今回



は、スタートメニューの「アクセサリ」の中にある
「Text Editor」を使おう。

エディターが開いたら、半角で次のように入
力しよう。

```
puts("hello, world")
```

「puts」は、英語で「置く」という意味で、「い
う」や「記入する」という意味もある。この後に続
く「()」の中のものを表示する命令だ。

「」(二重引用符、ダブルクオーテーション)で
囲まれている部分は文字列だね。これはなんでもいいんだけど、初めてのプログラミング言語を
使うときは「hello, world」を使うのが伝統にな
っている。

正しく入力ができたら、「ファイル(F)」メ
ニューから「保存(S)」を選んで保存する。名前は
「hello.rb」と入力(①)し、場所は「pi」(②)を
選ぼう。



保存したプログラムを実行するには、ターミナルを使う。タスクバーの「>_」のアイコンをクリックするか、スタートメニューの「アクセサリ」の中にある「LXTerminal」を選ぼう。

【ソースコード①】

```
class Sprite
  def say(word)
    puts(word)
  end
end
```

…「Sprite」(スプライト)の「class」(どのようなものか)を定義する
…「word」(単語)で与えられた文字列を「say」(いう)する方法を「def」(定義)する
…「puts」を使って単語を画面に表示する
…「say」の定義終わり
…「Sprite」の定義終わり

```
cat = Sprite.new          …「new」(新しい)スプライトをつくる、変数「cat」(ネコ)に入れる
cat.say("hello, world")  …ネコに「hello, world」といわせる(画面に表示する)
```



ターミナルが開いたら、「pi@raspberrypi:~ \$」の後の「█」に、次のように入力してから、「Enter」キーを押そう。すると、その下に「hello, world」と表示されるはずだ。

```
ruby hello.rb
```



こうならないときはエディターに戻り、打ち間
違いかないか確認しよう。文字で書くテキスト型
の言語ではスクラッチのようなブロック型の言
語と違い、一字一句正確に入力する必要がある。

スプライトのようなものをつくる

初期状態のルビーには、スクラッチのスライ
トやコスチューム、ステージのようなものがな
い。でも、自分でつくろうと思えばつくるこ
ができるんだ。エディターの「ファイル(F)」メ
ニューから「新規(N)」を選んで新しいウィンドウ
を開いたら、ソースコード①を打ち込もう。

…」の後ろの部分は説明なので入力しなくていいよ。

入力できたら、「cat.rb」という名前で保存する。そしてターミナルのウィンドウに戻り、次のように入力して「Enter」キーを押そう。

```
ruby cat.rb
```

```
pi@raspberrypi: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi:~ $ ruby hello.rb
hello, world
pi@raspberrypi:~ $ ruby cat.rb
hello, world
pi@raspberrypi:~ $
```

さっくと同じように「hello, world」と表示されたね。それにしても、ネコに一言いわせるだけなのに結構面倒だ。スクラッチには「言う」以外にもたくさんの命令があるけど、スクラッチのようなものをつくるのはどれだけ大変なんだ……。

そこで、ルビーには「gem」と呼ばれるライブラリ(スクラッチの拡張機能のようなもの)が用意されている。目的に合ったジェムを読み込むことで、いろいろな機能を組み合わせたプログラムをつくることができるんだ。

あるいは、ルビーの文法だけを使って、それを解釈して実行するためのしくみ(インタープリターやコンパイラー、トランスペイラー)を別に用意するという方法もある。今回は、そのようなものの1つ、スクラッチの見た目はそのままに、ブロックではなく、ルビーでプログラムを書けるようにした「Smalruby3」を使ってみよう。

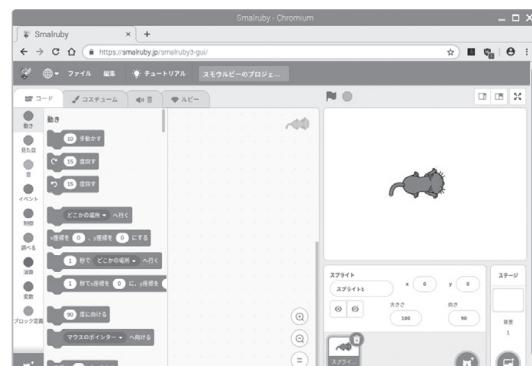
スモウルビー3でスクラッチのようにプログラミング

スモウルビー3は、高尾宏治さんたちが開発している、ブラウザ上で動くプログラミング環境だ。

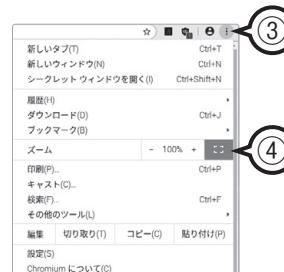
いつものように、スタートメニューの「インターネット」から「Chromium ウェブ・ブラウザ」を選ぶか、タスクバーの地球のアイコンをクリックして、クロミウムを起動しよう。クロミウムが起動したら、アドレスバーに

<https://smalruby.jp/smalruby3-gui/> と入力してから「Enter」キーを押すよ。

スモウルビー3の起動画面が開いたね。ほとんどスクラッチ3.0と同じだ。それもそのはず。スモウルビー3は、スクラッチ3.0をフォーク(リミックス)してつくられているんだ。



画面を広く使いたいので、全画面モードにしておこう。右上の「…」(3)をクリックして、「ズーム」の右端にある「[]」のアイコン(4)を選ぼう。全画面モードをやめたいときは「F11」キーを押してね。



ルビーでネコを動かす

では、ルビーのプログラムで、いつものようにネコ(スプライト1)を左右に動かしてみよう。「ルビー」のタブ(5)をクリックすると、テキストエディターの画面に切り替わるので、そこに右ページ上のソースコード⑥を半角で入力しよう。「.」(ドット)や「:」(コロン)などの記号にも注意してね。「_」(アンダースコア)はShiftキーを押しながら、「\」のキーを押すと入力できる。



【ソースコード⑧】

```

self.when(:flag_clicked) do ...緑の旗がクリックされたとき
  loop do ...ずっと
    move(10) ...10歩動かす
    bounce_if_on_edge ...もし端に着いたら、跳ね返る
  end ...ここまでを繰り返す
end ...ここまでを実行する

```

先頭の「self」は自分自身、つまり、この場合はネコのスプライトのこと。定義(def)していないのに「when()」などが使えるのは、すでにスマウルビー3の中で定義されているからだ。

緑の旗をクリックすると、ネコがステージを左右に往復するね。

動かないときは、打ち間違えているかもしれない。赤い「x」がついた行や、その前後をよく見て、間違いが見つかったら直そう。

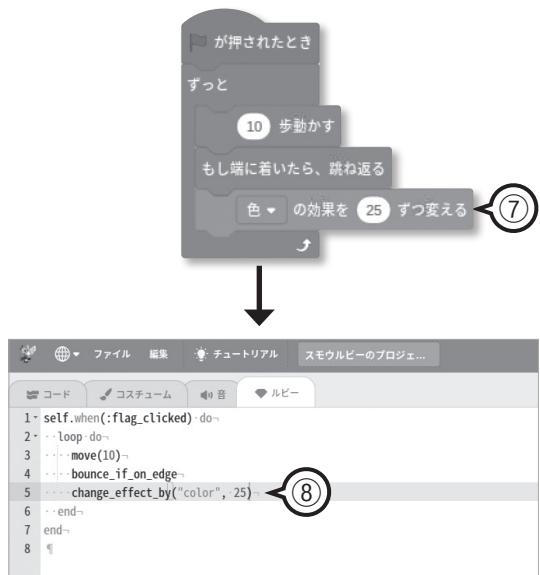
スクラッチとルビーのプログラムの比較

「コード」タブ(⑥)をクリックしてみよう。ブロックを組み立てていないのに、スクラッチのプログラムができている！



ここでスクラッチのプログラムを変えると、ルビーのプログラムはどうなるんだろう？ 試しに、「見た目」カテゴリーの「[色▼]」の効果を(25)ずつ変える」を「ずっと」の中に入れてから(⑦)、「ルビー」タブに切り替えてみよう。

「change_effect_by("color", 25)」が自動的に追加されているね(⑧)。これはとっても便利！



スクラッチのようなブロック型の言語でプログラミングを始めた人は、「それなら最初からスクラッチでプログラムした方が楽なんじゃない？」って思ったかもね。逆に、ルビーのようなテキスト型の言語で始めた人は、反対の感想を持つかも。

ブロック型にもテキスト型にも、よいところと悪いところ、向き不向きがある。スマウルビー3を使って「コード」と「ルビー」のタブを切り替えながら、考えてみてほしい。

つくったプログラムを保存するには、「ファイル」メニューから、「コンピューターに保存する」を選ぶ。保存されるファイル形式は「sb3」なので、スクラッチ3.0でも読み込める。それとは逆に、スクラッチ3.0でつくったプログラムをスマウルビー3で開くこともできるよ。試しに、先月号で紹介した感染症拡散シミュレーションをルビー言語で見てみよう(この記事のタイトル部分にある画面だ)。何か気づくことはあるかな？