



My Ruby tools for AtCoder

QWYNG(大澤広朗)

SmartHR LT会 May, 2024

AtCoderにRubyで参加する



- AtCoder: 競技プログラミングのコンテストを開催しているサービス
 - <https://atcoder.jp>
- Rubyで参加する際に利用している自作ツールの自慢をします
 - Rubyを使ったアルゴリズムの解説はしません

AtCoderに参加するときの流れ



1. サンプル入出力からテストを生成する
 2. 問題を解く
 3. デバックする
 4. テストを通す
 5. 提出
- 1と3で自作ツールを使っています

サンプル入出力からテストを生成する

- green_day
- https://github.com/QWYNG/green_day
 - 自作のスクレイピングツール
 - 問題の入出力からテストを自動生成してくれます
 - コンテスト開催中等クッキーが必要な場合にも対応しています

green_dayのデモ(1)



入力

入力は以下の形式で標準入力から与えられる。

`K X`

出力

総額が X 円以上である場合は `Yes` を、そうでない場合は `No` を出力せよ。

入力例 1 Copy

`2 900`

Copy

出力例 1 Copy

`Yes`

Copy

500円玉が2枚なので総額は1000円であり、これは $X = 900$ 円以上です。

abc150a

green_dayのデモ(2)



```
$ bundle exec green_day new abc150
```

```
abc150
├── A.rb
├── B.rb
├── C.rb
├── D.rb
├── E.rb
├── F.rb
└── spec
    ├── A_spec.rb
    ├── B_spec.rb
    ├── C_spec.rb
    ├── D_spec.rb
    ├── E_spec.rb
    └── F_spec.rb
```

green_dayのデモ(3)



```
RSpec.describe 'abc150/A.rb' do
  it 'test with "2 900\n"' do
    io = IO.popen('ruby abc150/A.rb', 'w+')
    io.puts("2 900\n")
    io.close_write
    expect(io.readlines.join).to eq("Yes\n")
  end

  it 'test with "1 501\n"' do
    io = IO.popen('ruby abc150/A.rb', 'w+')
    io.puts("1 501\n")
    io.close_write
    expect(io.readlines.join).to eq("No\n")
  end

  it 'test with "4 2000\n"' do
    io = IO.popen('ruby abc150/A.rb', 'w+')
    io.puts("4 2000\n")
    io.close_write
    expect(io.readlines.join).to eq("Yes\n")
  end
end
```

デバックスする(1)



テストで入出力をキャプチャしているのでREPLが動かない

```
> rspec abc150/spec/A_spec.rb
FFF

Failures:

  1) abc150/A.rb test with "2 900\n"
     Failure/Error: expect(io.readlines.join).to eq("Yes\n")
     ~~~~~ snip ~~~~~
     (compared using ==)

     Diff:
     @@ -1,9 +1,18 @@
     +
     +From: abc150/A.rb @ line 3 :
     +
     +  1: k, x = gets.split.map(&:to_i)
     +  2: check_complition = '予測変換もいい感じになってほしいなあ'
     + => 3: binding.irb
     +  4: puts k * 500 >= x ? 'Yes' : 'No'
     +Switch to inspect mode.
     +
     +  Yes
```

デバックスする(2)



- リモートプロセスのIRBを別のプロセスから操作することができるGemを作りました
 - <https://github.com/QWYNG/irb-remote>

irb-remoteのデモ(1)



```
3.2.2 ~/slide main +~ rspec abc150/spec/A_spec.rb
3.2.2 ~/slide main +~ irb-remote
Connected to remote session on druby://127.0.0.1:9876
From: abc150/A.rb @ line 4 :
1: require 'irb_remote'
2: k, x = gets.split.map(&:to_i)
3: check_complition = '予測変換もいい感じになってほしいな'
=> 4: binding.irb_remote
5: puts k * 500 >= x ? 'Yes' : 'No'
Switch to inspect mode.
irb-remote> [1, 2].each{|i|
irb-remote> i
irb-remote> end
[1, 2].each do |i|
  i
end

[1,
 2]
irb-remote> check_complition
check_complitionPress Option+d to read the full document
String < Object

Includes:
Comparable (from ruby core)
(from ruby core)
```

irb-remote

irb-remoteの先行研究



- dRubyを使ってオブジェクトをプロセス間でやりとりしています
- dRubyでREPLを行う発想は先人の方を参考にしたものです
 - <https://github.com/Mon-Ouie/pry-remote>
 - https://github.com/iguchi1124/irb_remote
- 今回はRelineを使いたかったので自作しました

まとめ



- 皆さんもぜひ使ってみてください