

Pythonのこの機能がRubyにも
欲しい！

unasuke

Ruby30周年記念イベント

2023-02-25

自己紹介

- Name: うなすけ
- Work: フリーランス
- Ruby歴: 8年くらい
- Kaigi on Rails オーガナイザー
- GitHub <https://github.com/unasuke>
- Mastodon <https://mstdn.unasuke.com/@unasuke>
- Twitter https://twitter.com/yu_suke1994



Pythonのこの機能がRubyにも欲しい！

とは？



Ruby Associationとは ▾

協賛募集 ▾

Ruby技術者認定試験

事業者認定制度 ▾

[Ruby Association](#) > [News一覧](#) > [Rubyアソシエーション開発助成金2022 公募開始](#)

Rubyアソシエーション 開発助成金2022 公募開始

Rubyアソシエーション開発助成金概要

Rubyアソシエーションは、Ruby処理系、ライブラリ及びフレームワーク等に関連した下記のテーマについて開発プロジェクトを募集し、助成金を交付します。よいアイデアがあれば、ぜひ提案を応募してください。ただし、この助成金の目的はRuby及び周辺環境の改善にあるため、特定目的のアプリケーションを開発するプロジェクトは対象外ですので、ご注意ください。

<https://www.ruby.or.jp/ja/news/20220823>

自分が今やっていること

RubyによるQUICプロトコルの他言語からの移植ならびに独自実装の作成

プロジェクト概要

2021年に標準化されたインターネットプロトコルであるQUICの利用は急速に広まっており、様々なプログラミング言語による実装が盛んである。中には複数の実装があるプログラミング言語も存在する。しかし現時点で公になっている、Rubyによる実装は存在していない。

このプロジェクトでは、最終的にRubyによるQUICプロトコルの実装を作成することを目指す。まずはその前段階として、Rubyに似たPythonによるQUIC実装である、aioquicをRubyに移植することで、QUIC実装の指針、知見を構築する。

応募者名

unasuke (Yusuke Nakamura)

<https://www.ruby.or.jp/ja/news/20221027>

自分が今やっていること

aiortc/aioquic

QUIC and HTTP/3 implementation in Python



 20

Contributors

 7

Issues

 18

Discussions

 1k

Stars

 179

Forks



<https://github.com/aiortc/aioquic>

自分が今やっていること

```
> tokei
```

```
=====
Language           Files      Lines      Code      Comments      Blanks
=====
Autoconf            1           4           4           0           0
C                   2          1025        751          128          146
CSS                 1           10           9           0           1
HTML                2           63           63           0           0
JSON                1           3            3           0           0
Makefile            1           20           10           6           4
Python              55         22918       18483        1230         3205
ReStructuredText    8           463          300           0           163
SVG                 1           72           72           0           0
Plain Text          2           5            0            5           0
TOML                1           2            2           0           0
=====
Total               75         24585       19697        1369         3519
=====
```

```
> |
```

まえおき

- Pythonはもっぱら読むがわ
- プロトコルの実装という分野においての意見
- Rubyにこの機能の輸入をしてほしいか、と言われると……

Rubyにこの機能を入れてほしい（ではない！）

- 既存資産との兼ね合いとか
- 文法的に厳しいとか
- 「みんなちがって、みんないい。」的な気持ちで

その1 Enum

```
from enum import Enum
class Color(Enum):
    RED = 0
    GREEN = 1
    BLUE = 2
```

EnumはRubyでも実現できるのでは？

```
class Color
  RED = 0
  GREEN = 1
  BLUE = 2
end
```

ちやうねん

EnumはRubyでも実現できるのでは？

```
from enum import Enum
class Color(Enum):
    RED = 0
    GREEN = 1
    BLUE = 2

foo = Color.RED
print(foo) # => "Color.RED"
```

Pythonだところ

EnumはRubyでも実現できるのでは？

```
class Color
  RED = 0
  GREEN = 1
  BLUE = 2
end
foo = Color::RED
puts(foo) # => "0"
```

Rubyだとう

その2 bytes

```
foo = b"\x41\x42\x43" # ABC
len(foo) # => 3
```

その2 bytes (Rubyの場合)

```
foo = "\xe3\x81\x82"  
foo.force_encoding(Encoding::ASCII_8BIT)  
bar = "あ"  
foo == bar # => false
```

RubyだとStringになり、Encodingを持つ

その2 bytes (RubyのIO::Bufferの場合)

```
a_ascii = "\xe3\x81\x82"  
a_ascii.force_encoding(Encoding::ASCII_8BIT)  
a_utf8 = "あ"  
foo = IO::Buffer.for(a_ascii)  
bar = IO::Buffer.for(a_utf8)  
foo == bar # => true
```

IO::Bufferならバイト列になる

その3 tuple

```
foo = ("A", 0x41)
print(foo) # => '("A", 65)'
```

その3 tuple (Rubyの場合)

```
foo = ["A", 0x41]
pp foo # => ["A", 65]
```

配列にならざるをえない

その4 @dataclass (というよりはdecorator)

```
@dataclass
class Dog:
    name: str
    age: int
```

その4 @dataclass (Rubyの場合)

```
# Struct
Dog = Struct.new(:name, :age)

# Data
Dog = Data.define(:name, :age)
```

StructとかDataになる

その4 @dataclass (というよりはdecorator)

```
@foo
class Dog: pass

# 上と等価
Dog = foo(Dog)
```

Classや関数をwrapできる

逆にRubyのこの機能はうれしい

- 後置if/unless
- de facto standardなパッケージマネージャ(gem/bundler)
- debug.gem (色)

おしまい