

# Python 版 vitocha : vitothon

アルティウスリンク（旧 KDDI エボルバ） 松岡主馬

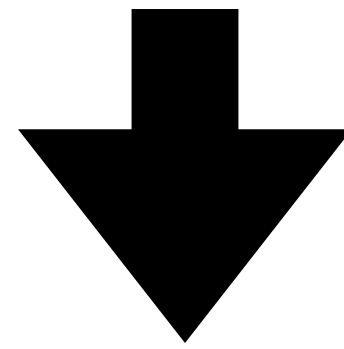
# 自己紹介

- 松岡主馬（かずま）
- 経歴
  - 2020 大学院休学
  - 2021 HashPort 入社 -> HashBank
  - 2022 復学 -> なんとか卒業
  - 2023 HashBank 退社 -> KDDI エボルバ入社  
(2023/9 アルティウスリンクに統合)
- 趣味：ゲーム (YouTube 再開？)
- 参加した経緯：就活



VITOTHON とは

VITOTHON



VITOCHA + Python  
(Python 版 VITOCHA)

Q. なぜ Python で作り直したのか

2018 年春

これが VITOCHA です。

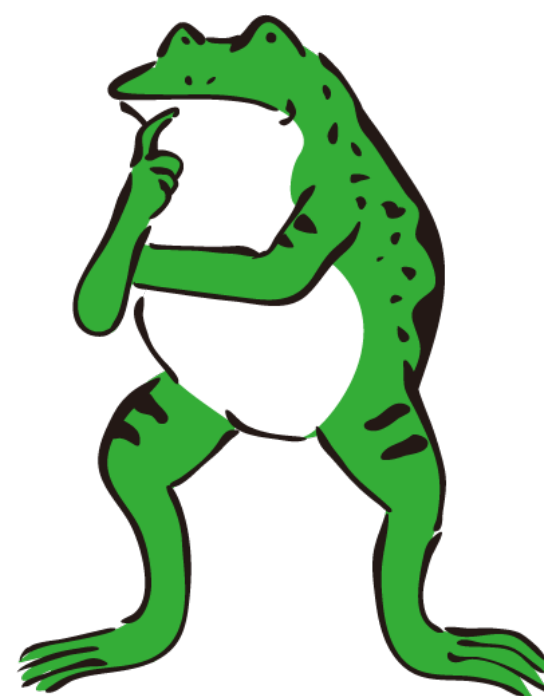


卒論の実験で使えるかもしれない。

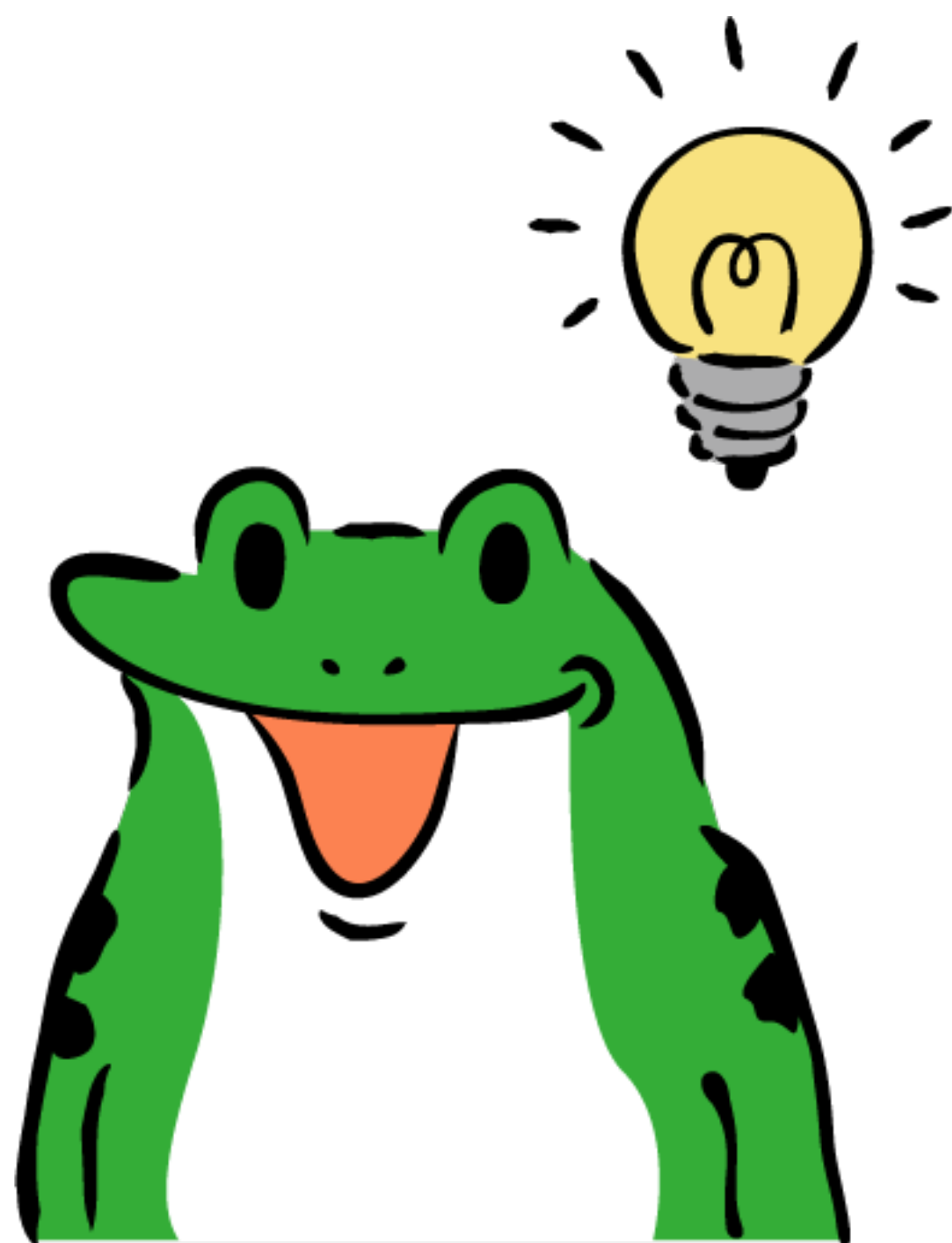
先生の説明を聞いても中身はわからない。

機能追加もしたい。

Python 覚えたばかりで Ruby 使うのは・・・







分解したら作れた。



# VITOCHA

FreeBSD

- jail
- VIMAGE

Ruby

- Shell クラス

# VITOCHA

## Example

```
=====
|-- bridge.rb
|-- router.rb
|-- server.rb
|-- equipment.rb
=====
|-- mkrouter
|-- mkserver
=====
|-- nwdiag2jail.rb
=====
|-- shcommand.rb
|-- unjail.rb
|-- vitocha.rb
=====
|-- simulator.rb
```

機器作成

jail 作成

nwdiag

jail 設定

実行例

```
router0=Router.new("router0")
router0.connect("epair0b")
```

```
./mkrouter router0
```

```
tomocha=Operator.new
ifconfig("epair0a up")
```

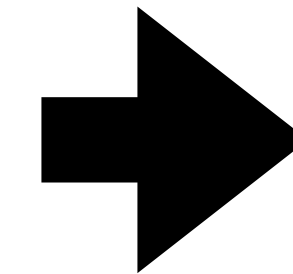
# VITOTHON

```
├─ README.md
├─ nwdiag      # ネットワーク図描画用フォルダ
├─ shell       # jail 作成スクリプト
├─ src         # VITOTHON 本体
└─ test        # VITOTHON の使用例
```

# VITOCHA -> VITOTHON

```
=====
|-- bridge.rb
|-- router.rb
|-- server.rb
|-- equipment.rb
=====
```

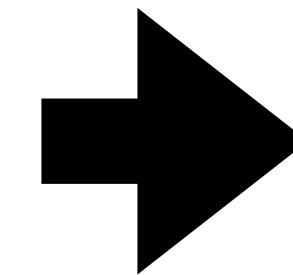
機器作成



src/equipment.py に集約

```
=====
|-- mkrouter
|-- mkserver
=====
```

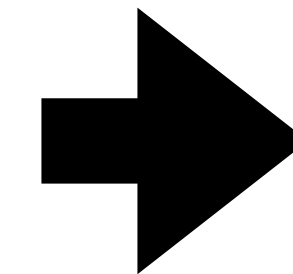
jail 作成



shell フォルダに集約

```
=====
|-- nwdiag2jail.rb
=====
```

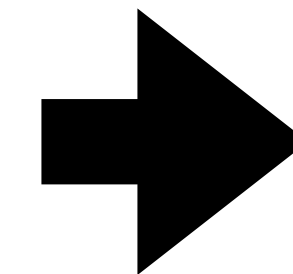
nwdiag



nwdiag フォルダに集約

```
=====
|-- shcommand.rb
|-- unjail.rb
|-- vitocha.rb
=====
```

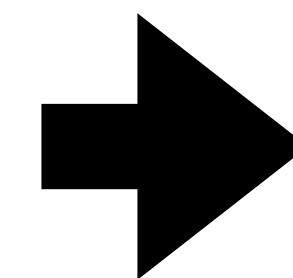
jail 設定



src フォルダに集約

```
=====
|-- simulator.rb
=====
```

実行例



test フォルダに集約

# そうだ **Ruby** の **Shell** クラス使おう

- こんな風にしておいて

```
Shell.def_system_command("ifconfig_org", path = "/sbin/ifconfig")
def ifconfig(param)
  sh=Shell.new
  sh.transact{
    ifconfig_org(*param.split)
  }
end
```

VITOTHON では subprocess で代用

```
def ifconfig(param):
  arg = param.split()
  arg.insert(0, "ifconfig")
  subprocess.run(arg)
```

# 機器はオブジェクトにしてみよう

- こうしておくと、

```
class Router
  def initialize(jailname)
    ...
    jail(jailname)
    jexec(jailname, 'ifconfig lo0 127.0.0.1/24 up')
  end

  def assignip(epair, ip, mask)
    ...
  end
end
```

VITOTHON でも同じように実装

```
class Router(Equipment):
    def __init__(self, jailname):
        super().__init__(jailname)
        jexec(jailname, "sysctl -w net.inet.ip.forwarding=1")
```

# evalな操作を隠蔽してしまいたい

- そうだ仮想オペレータにやらしてもらおう
- 機器の設定をやらしてもらおう (eval操作の隠蔽化)
- 配線管理もまかせちゃおう

```
class Operator

  def setupserver(jailname)
    eval("$#{jailname}=Server.new('#{jailname}')" )
  end

  def createpair
    ...
  end
```

VITOTHON でも同じように実装

```
class Operator:
    def __init__(self):
        ...

    def setupserver(self, jailname):
        exec("{0} = Server('{0}').format(jailname), globals())
```



```
# VITOCHA
```

```
tomocha.createpair
```

```
router0.connect("epair1a")
```

```
bridge0.connect("epair1b")
```

```
router0.assignip("epair1a","172.18.128.254","255.255.255.0")
```

```
tomocha.register("epair1a","router0","172.18.128.254","255.255.255.0")
```

```
tomocha.register("epair1b","bridge0")
```

```
bridge0.up("epair1b")
```

```
# VITOTHON
```

```
epaira, epairb = op.createpair()
```

```
op.connect("router01", epaira)
```

```
op.connect("bridge00", epairb)
```

```
op.assignip("router01", epaira, "192.168.100.1", "255.255.255.0")
```

```
op.up("router01", epaira)
```

```
op.up("bridge00", epairb)
```

- 仮想インターネットの構築 ○
- DNS ルートサーバを作る ▲ アプライアンスに含まれていない (実証済み)
- IX を作る ▲ ブリッジは実装済み
- Tier1 BGP4 ネットワークを作る ○ ただしポリシー考慮はしていない
- Tier2 .. ▲ 未実証
- OSPF ネットワークを作る ○
- サーバセグメントを作る ○
- WWW.WINE.NOM を立ち上げる ▲ 新しい VITOTHON では未実証
- ネットワーク図を nwdiag で作成する ▲ 手動なら可能



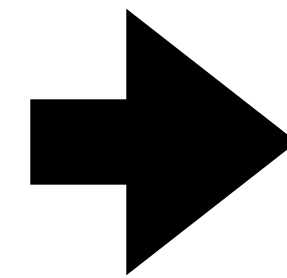
1) 卒論

2) ネットワークルーティング攻撃に対する  
ブロックチェーンシミュレーション

3) Blockchain and DHT Based Lookup System  
Aiming for Alternative DNS

成果物を公開する

ドキュメントを残す



GitHub

README, Zenn



# 今後の VITOTHON

- 仮想スイッチを導入
- nwdiag 自動化
- 各種サーバ用 flavour を作成
- GitHub にて共同開発 or Issue の募集

余談



子供ができました。



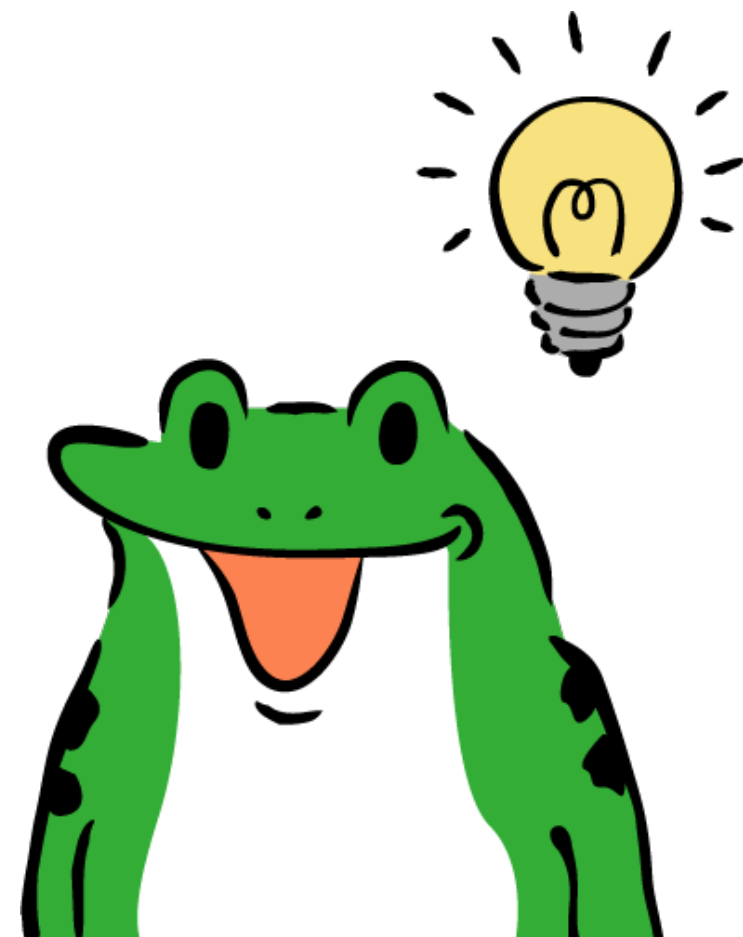


目標設定して。



# 目標設定

1. クライアントへの貢献
2. 技術力の向上・自己研鑽
3. チームマネジメント、育成への協力等



VITOTHON アプライアンス化したら  
2, 3 がうめられる！